Computers and Electronics in Agriculture 175 (2020) 105397

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

An obstacle separation method for robotic picking of fruits in clusters

Ya Xiong, Yuanyue Ge*, Pål Johan From

Faculty of Science and Technology, Norwegian University of Life Sciences, Ås, Norway

ARTICLE INFO

Keywords:
Agricultural robotics
Obstacle separation
Strawberry-harvesting robots

ABSTRACT

Selectively picking a target fruit surrounded by obstacles is one of the major challenges for fruit harvesting robots. Different from traditional obstacle avoidance methods, this paper presents an active obstacle separation strategy that combines push and drag motions based on 3D visual perception to separate obstacles from the target. We define a region of interest 3D point cloud with a number of sub-blocks around the target to determine the presence or absence of obstacles and generate the separation paths accordingly. A linear push is used to clear the obstacles from the area below the target, while a zig-zag push that contains several linear motions is proposed to push aside more dense obstacles. The zig-zag push can generate multi-directional pushes and the side-to-side motion can break the static contact force between the target and obstacles, thus helping the gripper to receive a target in more complex situations. Moreover, we propose a novel drag operation to address the issue of mis-capturing obstacles located above the target, in which the gripper drags the target to a place with fewer obstacles and then pushes back to move the obstacles aside for further detachment. Furthermore, an image processing pipeline consisting of color thresholding, object detection using deep learning and point cloud operation, is developed to implement the proposed method on a newly developed harvesting robot. Field tests show that the proposed method can improve the picking performance substantially. This method helps to enable complex clusters of fruits to be harvested with a higher success rate than conventional methods.

1. Introduction

Fruit production that requires selective harvesting is heavily reliant on human labour (Xiong et al., 2019). This is applicable to crops such as strawberries, sweet peppers, tomatoes, cucumbers, etc. Labour represents the largest cost and also a large operational uncertainty for fruit growers (Yamamoto et al., 2014). Therefore, several attempts have been made to develop a robotic solution for selective harvesting of fruits. Some fruits, such as strawberries and tomatoes, tend to grow in clusters. This makes it difficult to identify and pick individual ripe fruit without damaging or accidentally picking unripe fruits (Xiong et al., 2020; Yamamoto et al., 2014). Harvesting fruits that grow in clusters or are surrounded by obstacles, such as branches and/or leaves, while leaving the other fruits to remain undamaged on the plant, is one of the primary challenges for fruit harvesting systems (Xiong et al., 2020; Yaguchi et al., 2016). The surrounding fruits, leaves, stems and other obstacles are often difficult to separate from the target, both in terms of detection and in manipulation.

In agricultural robotics field, many researchers try to avoid obstacles in both vision and manipulation. To avoid occlusions in sweet pepper picking, a "3D-move-to-see" method was proposed to find the best view with fewer occlusions (Lehnert et al., 2018). To avoid obstacles, a method for cucumber picking was developed that uses a search algorithm to explore the search space for a feasible trajectory, in which each step of the trajectory is checked by a collision detector (Van Henten et al., 2002). A grape vine pruning robot employed a similar randomized path planner, namely rapidly exploring random tree (RRT), in which a collision detector is used to check the robot's self-collision or collision with other obstacles (Botterill et al., 2017). Also, an improved RRT algorithm combining target gravity and genetic algorithm was proposed to generate a collision-free path for litchi-picking robot (Cao et al., 2019). Most of the methods found in the literature are passive obstacle avoidance methods, in which the aim is to avoid existing obstacles without changing the environment. However, obstacles are not always avoidable, especially when picking small-size fruits in clusters, where the obstacles may be extremely close to the targets. For example, as shown in Fig. 1(a), a target strawberry is completely surrounded by obstacles, in which the gripper is difficult to find a way to avoid all obstacles.

Our previous work presented a gripper for strawberry harvesting that can open fingers to capture a target from below (Xiong et al., 2019). Without moving the obstacles out of the way, obstacles may

* Corresponding author.
E-mail address: yuge@nmbu.no (Y. Ge).

https://doi.org/10.1016/j.compag.2020.105397
Received 24 November 2019; Received in revised form 26 February 2020; Accepted 30 March 2020
Available online 15 June 2020

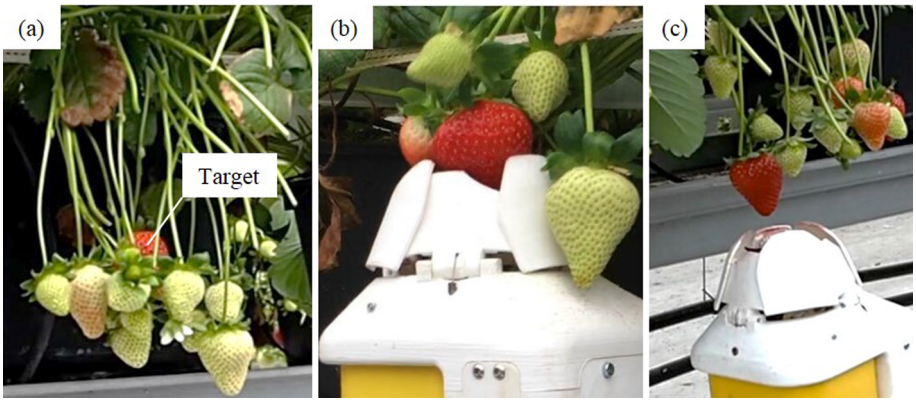

Fig. 1. Obstacles around the target ripe strawberries make the selective picking extremely hard: (a) a target is fully surrounded by obstacles; (b) a target strawberry cannot fall down to the gripper; (c) the obstacles above the ripe strawberry may be captured by the gripper with the target.

prevent the gripper from capturing the target and may also be swallowed with the target if they are located close to the target. Similar problems occur when approaching the fruit from other angles. To solve this issue, in a later work (Xiong et al., 2020), we proposed to use a single linear push operation to actively push aside the obstacles below the target based on the obstacle sensing from a 3D camera. We found that pushing obstacles aside, rather than simply avoiding them makes it possible to pick fruit that would otherwise be inaccessible to the robot. However, a single linear push may be insufficient for obstacles from multi-direction with respect to the target, such as the case in Fig. 1(a), since the linear push moves towards only one direction. Moreover, the obstacles may be adjacent together that can not be separated during the single push. Furthermore, as shown in Fig. 1(b), when the obstacles connect to the target at the same height, the gripper may not be able to swallow the whole target but push it up due to the static contact force between the target and obstacles. In addition to that, as shown in Fig. 1(c), one frequent failure is that the gripper may capture obstacles above the target when it moves up to detach the target, which has not been addressed in the previous work.

In the field of robotic manipulation, most studies focus on obstacle avoidance. Nevertheless, we found some studies working on obstacle separation for simple situations. For a warehouse picking application on desk 2D environment, two linear pushing policies were proposed to separate the rigid obstacles during the way of the gripper reaching a target bin (Danielczuk et al., 2018). All the objects were places on the 2D surface without stacking, which is very simple compared to natural growing plants. Another work used Learning from Demonstration (LfD) algorithm for the same application that involves a pushing action (Laskey et al., 2016), which is slow in operation. For a similar situation, researchers proposed to use physical engine to calculate the dynamics to predict the object locations for motion planning, which also involves pushing motions (Kitaev et al., 2015; Moll et al., 2017; Dogar et al., 2013). Reinforcement learning was also used to train a robot to rearrange objects on a desk using pushing method to make them sparse for individual grasping (Zeng et al., 2018). However, all these methods are targeted at simple environment where several rigid objects were placed on a 2D desk surface without stacking, so it is a 2D motion-planning problem. Without stacking, the vision system can easily track the target and obstacles, which facilitates the closed-loop vision guided control. Unfortunately, in the agricultural environment, for example strawberry plants, fruits are located in 3D within diverse and unconstrained environments. Selectively picking a ripe fruit in clusters requires 3D motion planning to separate the target from obstacles if using obstacle separation methods. The occlusions make the vision system difficult to track the changes of the objects. Also, the flexible peduncles, deformable fruits and many other crop variations make the separation operations extremely difficult and the dynamics of these objects are difficult to calculate and predict. Moreover, the operation speed of the learning or physical engine based methods seems very slow, which may not be suitable for fruit harvesting.

In this paper, we provide the improvements to our previously proposed active obstacle separation method (Xiong et al., 2020). The proposed method might be also applicable to harvest other fruits, such as tomatoes and cucumbers. The main contributions are as follows:

1. We define a more sophisticated region of interest 3D point cloud around the target to generate higher resolution of separation paths.
2. We extend the single linear pushing policy by adding a zig-zag push for both horizontal and upward directions to deal with more complex situation.
3. A novel in-hand drag operation is proposed to avoid mis-capturing of upper obstacles, which includes avoiding obstacles and pushing obstacles.
4. To implement the method, we propose an image processing pipeline that combines 3D color thresholding, 2D object detection using deep learning and point cloud operation to output both target position and obstacle information.

2. Methods

2.1. Region of interest

The obstacle separation paths are generated according to the visual perception of the obstacle information around the target. We select a region of interest (RoI) area around the target to determine the presence of the obstacles and calculate the separation paths based on the distribution and number of the obstacles. The RoI comprises a volume of 3D point cloud that contains the target fruit and potentially one or more obstacles. As shown in Fig. 2, the RoI area is divided into four horizontal layers: a top layer 1, an upper-central layer 2, a lower-central layer 3 and a bottom layer 4.

As the top view is shown in Fig. 2, each layer of the RoI is further segmented into nine cuboid blocks. On each layer, the blocks are arranged in a 3x3 grid that has its center at the horizontal midpoint of the target strawberry such that the central block $\boldsymbol{C}_C$ encompasses the position of the target strawberry in the xy plane. In the top view, the length and width of the outside eight blocks are equal to the central block. In the front view and left side view, the heights of layer 1 and layer 4 are equal to one and two of the height of the target block, respectively. The gripper moves up to separate the obstacles around the target in the central layer. The obstacle distribution in the central layer may vary along the height direction. To get a higher resolution of motions, we break the motions in central layer into two steps, thus having layer 2 and layer 3. Particularly, the central block of layer 1 is shorter than other blocks in the same layer, 80% of other blocks. This is because the object segmentation method does not include the green

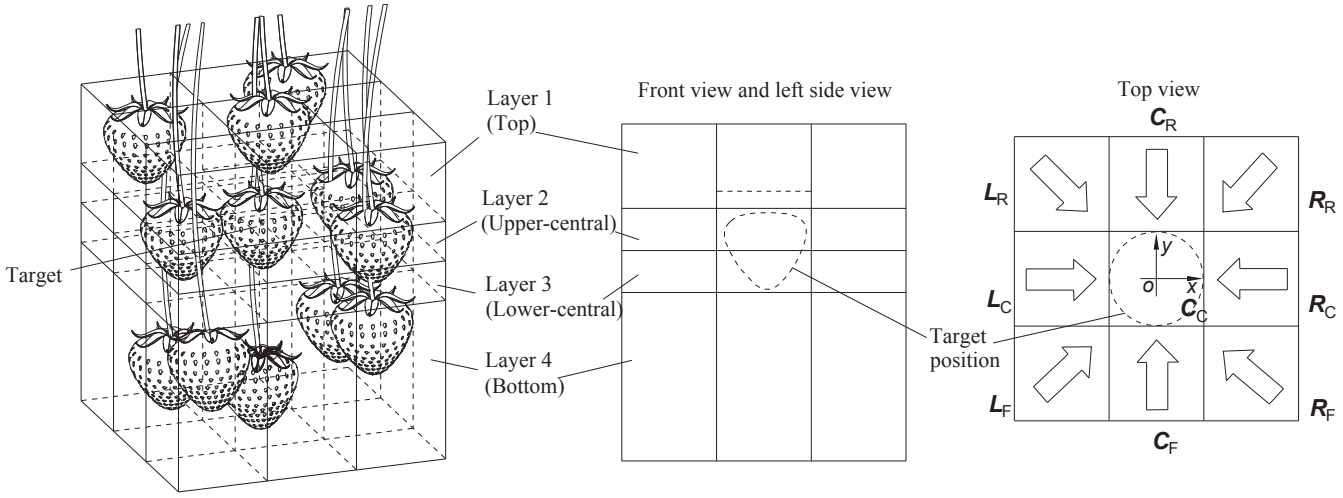

Fig. 2. Region of interest area around the target to determine the presence of obstacles.

calyx. To avoid the calyx being detected as an obstacle, we leave the bottom of the central block in layer 1 blank.

To generate the separation paths, each block is assigned a horizontal vector representative of the direction from the block to the central block C_C. The direction of the vector is determined by the position of the block so that all vectors are directed from the center of the corresponding block towards the center of the central block C_C. We use the number of points N in the block of point cloud to determine whether there are obstacles present in the block or not. In this paper, the thresholds of N for layer 1, layer 2/3 and layer 4 are 200, 100 and 300, respectively, using the resolution of 1280 × 720 of the RGB-D camera.

The gripper is instructed to operate in three distinct stages. As the gripper is picking from below, during the first stage, the gripper moves obstacles horizontally within the bottom layer. During the second stage, the device moves up to swallow the target and separates the obstacles within the central layers. During the third stage, the gripper may drag the target into a picking position with fewer obstacles if the central block C_C in the top layer is occupied. The detailed separation policies will be elaborated in the below sections.

2.2. Horizontal push in the bottom layer

The first stage is to separate the obstacles horizontally under the target in layer 4. Our previous work proposed a single linear push to move the obstacles out of the way (Xiong et al., 2020), which shows promising results in the situation with fewer obstacles but struggles in more complex situation. A single push means that the gripper linearly pushes the obstacles aside once, starting from the region with fewer obstacles. Inspired from our daily experience that a shaking operation can help insert an object into a target place surrounding with obstacles, we add a zig-zag pushing policy to move the obstacles side to side for more complex situation. A zig-zag push is a motion where the gripper uses a zig-zag movement that contains several linear motions to push the obstacles side to side. This motion can not only move the obstacles out of the way, but also break the static contact force, such as shaking to insert a key. However, a single push is generally faster than the zig-zag push and can reduce the likelihood of the damage to the fruit. Therefore, we only use the zig-zag push in more complex situation, depending on the number and distribution of the obstacles in layer 4.

2.2.1. Single push

As shown in Fig. 3(a), if an obstacle is located below the target (layer 4), the gripper may capture the obstacles if it moves up straightly to enclose the target. In this case, the gripper can use a single push operation to push aside the obstacle (to the right in the figure) before swallowing the target (Fig. 3(b) and (c)).

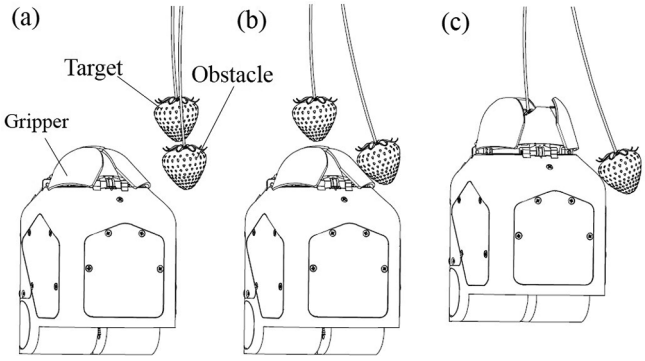

Fig. 3. Singe push to move the obstacles under the target in layer 4.

Since the gripper size is limited, a single push operation makes it easier to move a few number of adjacent obstacles out of the way, but hard to separate many sparsely distributed obstacles. Therefore, ignoring the central block, we use the number of blocks n_h within the largest group of adjacent unoccupied blocks (no obstacles) to determine whether to use a single push or a zig-zag operation. As shown in Fig. 4(a), ignoring the central block, filled arrows in the blocks mean that the blocks are occupied with obstacles, while the blank arrows represent unoccupied blocks. In this case, the n_h is 5 and greater than a

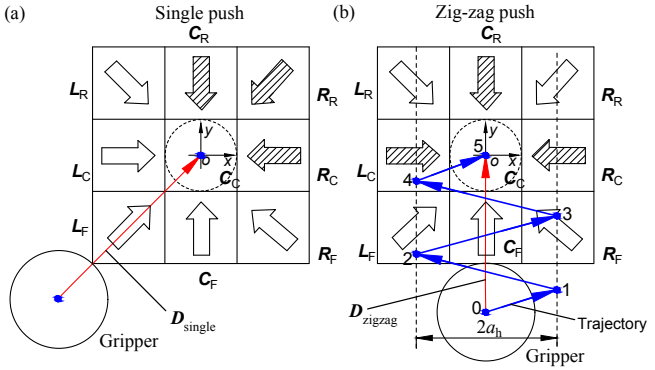

Fig. 4. Diagram of the calculation of the horizontal push: (a) single push, where the red arrow is the pushing direction; (b) zig-zag push, where the red arrow shows the overall direction and the blue arrows are the zig-zag push paths. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

predetermined threshold t_h (using 4 in this paper), so a single push operation is appropriate to push the obstacles aside. As the pushing operation moves towards the obstacles, the direction of the single push operation for the gripper is calculated based on the positions of the occupied blocks according to the following equation:

$$\boldsymbol{D}_{\text{single}} = -r\Sigma_1^n \boldsymbol{O}_i / |\Sigma_1^n \boldsymbol{O}_i|, \quad n_h >= t_h \tag{1}$$

where $\boldsymbol{O}_i$ is the vector of the i^{th} occupied block within the largest group of adjacent occupied blocks and n is the total number of blocks within the largest group of adjacent occupied blocks. The parameter r is used to scale the $\boldsymbol{D}_{\text{single}}$ norm, which should guarantee that the gripper pushes from the outside of the blocks (50 mm is used for the current system).

The red arrow in Fig. 4(a) shows the calculated pushing direction for the single push operation. It can be seen that the gripper moves from the center of the unoccupied blocks to the center of the occupied blocks, such that the gripper has the highest possibility to move all the obstacles out of the way. If only the central block $\boldsymbol{C}_{\text{C}}$ is occupied, then $\boldsymbol{D}_{\text{single}} = 0$. In this situation, the direction in which the gripper must move in order to push the obstacles is instead determined by calculating the shortest path from the current location of the gripper to the center of the central block $\boldsymbol{C}_{\text{C}}$. If no obstacles are detected in the blocks, the gripper has no pushing action at this stage and moves up straightly from the below.

2.2.2. Zig-zag push

Ignoring the central block, if the number n_h of the largest group of adjacent unoccupied blocks comprises fewer than the threshold number t_h of blocks, the method determines that a horizontal zig-zag push operation is appropriate. Fig. 4(b) shows a path calculation example where a zig-zag operation is selected to push the obstacles side to side. The red arrow is the overall direction of the operation, while the blue arrows are the zig-zag paths. Since the zig-zag operation involves three directions of movement (forward, left and right), the gripper can push the three directions of obstacles out of the way.

Different from the single push, the overall direction of the zig-zag push operation is calculated based on the positions of the unoccupied blocks according to the following equation:

$$\boldsymbol{D}_{\text{zigzag}} = r\Sigma_1^m \boldsymbol{U}_j / |\Sigma_1^m \boldsymbol{U}_j|, \quad n_h < t_h \tag{2}$$

where $\boldsymbol{U}_j$ is the vector of the j^{th} unoccupied block within the largest group of adjacent unoccupied blocks and m is the total number of blocks within the largest group of adjacent unoccupied blocks. During a horizontal zig-zag push operation, the device moves in the xy plane, wherein the resultant vector of the zig-zag motion is equal to $\boldsymbol{D}_{\text{zigzag}}$ and the amplitude a_h and number of pushes n_{hp} of the zig-zag motion are determined according to the specific picking scenario. For example, the effectiveness of the values may be affected by the peduncle length, fruit weight or the damping ratio of the fruit, which are difficult to calculate. Based on some tests in the farm, in the current system, we tune the a_h and n_{hp} to fix values of 20 mm and 5, respectively.

In a scenario where all of the blocks, except for the central block $\boldsymbol{C}_{\text{C}}$, are occupied, resulting in a value of $\boldsymbol{D}_{\text{zigzag}} = 0$, the $\boldsymbol{D}_{\text{zigzag}}$ is instead determined by calculating the shortest path from the current location of the gripper to the center of the central block $\boldsymbol{C}_{\text{C}}$.

2.3. Upward zig-zag push in the central layers

After finishing the separation in layer 4, the gripper moves up to enclose the target in layer 3 and 2. In the previous work, we introduced a method that moves the gripper with an offset to the central position of the target for swallowing a target with a connected berry (Xiong et al., 2020). This method can help to avoid swallowing the other connected obstacle. However, we observed that the target or the obstacles might not fall down, but are moved up by the gripper due to the static contact force. Similar to horizontal push, to break the static force, we add an upward zig-zag operation to help the gripper swallow the target and also make it easier for the obstacles to fall on the outside of the fingers.

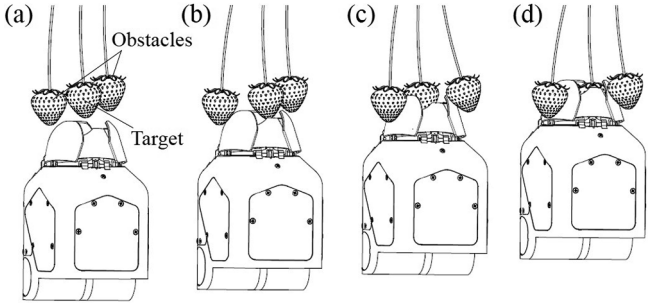

Fig. 5. Upward zig-zag operation: when moving upward, the gripper moves to the left and right to push aside two sides of obstacles.

As shown in Fig. 5, the upward zig-zag push operation comprises the movement of the gripper in a principally vertical direction towards the target fruit and a side-to-side movement to clear obstacles. The vertical direction passes through the center of the target. Similar to the horizontal push, we calculate the direction $\boldsymbol{D}_{\text{u\_zigzag}}$ of the upward push in xy plane based on the number of blocks n_u within the largest group of adjacent unoccupied blocks. If n_u is greater than the threshold t_h, the situation is the same to the single push in layer 4, so the direction $\boldsymbol{D}_{\text{u\_zigzag}}$ can be calculated according to the occupied blocks.

$$\boldsymbol{D}_{\text{u\_zigzag}} = -a_u\Sigma_1^n \boldsymbol{O}_i / |\Sigma_1^n \boldsymbol{O}_i|, \quad n_u >= t_h \tag{3}$$

where a_u is the parameter used to scale the $\boldsymbol{D}_{\text{u\_zigzag}}$ norm (5 mm is used for the current system). If the n_u is fewer than the threshold t_h, as shown in Fig. 6(a), the calculation then uses the unoccupied blocks, which is similar to the calculation of the zig-zag push in layer 4. The calculation can be concluded as following equation:

$$\begin{cases} \boldsymbol{M} \cdot \boldsymbol{D}_{\text{u\_zigzag}} = 0, \quad n_u < t_h \\ \boldsymbol{M} = \Sigma_1^m \boldsymbol{U}_j \\ |\boldsymbol{D}_{\text{u\_zigzag}}| = a_u \end{cases} \tag{4}$$

where $\boldsymbol{M}$ is the intermediate vector to calculate $\boldsymbol{D}_{\text{u\_zigzag}}$. Therefore, for the case in Fig. 6(a), the gripper moves along with $\boldsymbol{D}_{\text{u\_zigzag}}$ and $-\boldsymbol{D}_{\text{u\_zigzag}}$ to push aside the two sides of obstacles. The front view in Fig. 6(b) shows the gripper moves gradually at left or right intermediate points to pass over layer 3 and 2. We set the number of pushes n_{up} in each layer to 5 in the current system.

2.4. In-hand drag operation in the top layer

If an obstacle is located above the target (layer 1), such as the case shown in Fig. 7(a), the gripper may swallow or damage the obstacles when moving upward to capture the target strawberry. Furthermore,

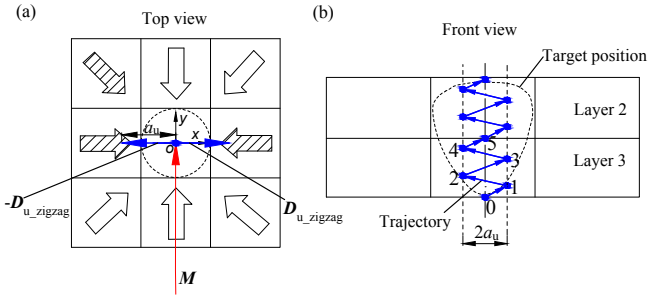

Fig. 6. Diagram of the calculation of the upward zig-zag push: (a) top view shows the calculation of the principle push direction; (b) front view shows the vertical paths (blue arrows). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

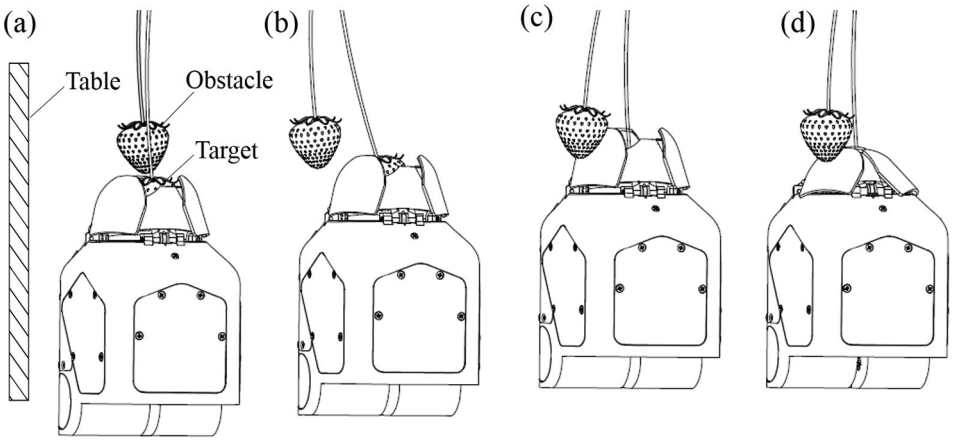

Fig. 7. Drag operation to avoid capturing the obstacles: an upward drag step moves the target to an area that contains fewer obstacles ((a) and (b)); an upward push-back step pushes the upper obstacles aside (c) before closing the fingers (d).

the obstacles may stop the fingers closing thus resulting in cutting failure of the target peduncle. To solve this problem, we propose an in-hand drag operation, which is opposite to the push operation as used in other layers. The drag operation allows the gripper to pick the target fruit without capturing unwanted obstacles. The operation comprises an upward drag step to move the target to an area that contains fewer obstacles (Fig. 7)) and an upward push-back step that pushes the upper obstacles aside (Fig. 7(c)) before closing the fingers. The push-back step is necessary because when at the drag position (Fig. 7(b)), the peduncle is inclined such that the fruit is difficult to fall due to the static force and easily damaged when the gripper moves up further towards a cutting position.

The drag operation is performed only when there are obstacles in the central block $\boldsymbol{C}_C$ of the top layer. If the $\boldsymbol{C}_C$ is unoccupied, the gripper moves directly upwards to pick the target strawberry. Fig. 8 shows the diagram of the calculation method of the drag operation with corresponding to the example in Fig. 7. As shown in Fig. 8(a), to avoid the collision between the gripper and the table, the three blocks $\boldsymbol{L}_R$, $\boldsymbol{C}_R$, $\boldsymbol{R}_R$ that are close to the table are skipped for the calculation of the drag direction. Then the drag direction $\boldsymbol{D}_{drag}$ in the *xy* plane can be determined according to the following equation:

$$\boldsymbol{D}_{drag} = l\Sigma_1^m \boldsymbol{U}_j / |\Sigma_1^m \boldsymbol{U}_j| \quad (5)$$

where $\boldsymbol{U}_j$ is the vector of the j^{th} unoccupied block within the largest group of adjacent unoccupied blocks. The blocks used for calculation are $\boldsymbol{L}_C$, $\boldsymbol{L}_F$, $\boldsymbol{C}_F$, $\boldsymbol{R}_F$, $\boldsymbol{R}_C$. The parameter m is the total number of blocks within the largest group of adjacent unoccupied blocks. The norm of $\boldsymbol{D}_{drag}$ is scaled to l (50 mm is used in the current system). If all blocks are occupied by obstacles, the drag direction is aligned to $\boldsymbol{C}_F$, where there are fewer obstacles in general. Fig. 8(b) shows the drag and push-back steps, wherein the drag and push-back operations moves up the same height in the vertical direction.

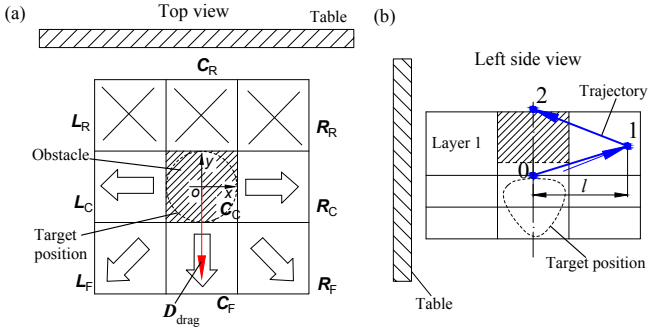

Fig. 8. Diagram of the calculation of the drag operation: (a) a top view shows the calculation of the drag direction $\boldsymbol{D}_{drag}$ in *xy* plane; (b) a left side view shows the drag and push-back paths (blue arrows).

3. Experiments

3.1. Image processing

The image processing includes the detection and localization of ripe strawberries and also the determination of obstacles within the RoI for each target. An RGB-D camera (D435; Intel, USA) was used for image acquisition. The image processing pipeline contains three steps: (1) 3D color thresholding to remove noise points from the background, (2) object detection and localization using deep learning based on our previous work (Ge et al., 2019) and (3) obstacle calculation.

Fig. 9(a) shows the original point cloud, where some pieces of points from the table (silver) and irrigation tubes (black) are around the strawberries. In fact, the table and irrigation tubes are behind the berries at a distance of about 150 mm. The inaccurate depth sensing results in some of the points connecting to the front berries, which may be regarded as obstacles. To avoid this influence, the first step is to remove the adjacent noise points (silver and black) by using hue, saturation and intensity (HSI) color thresholding, as the result is shown in Fig. 9(b). This step is performed in point cloud using the *jsk_pcl_ros* ROS package. The second step is the detection and localization of the ripe strawberries. As shown in Fig. 9(c), we use an instance segmentation convolutional neural network Mask R-CNN to identify and segment the objects in pixel level. Through the network, several masks are created for the ripe strawberries, in which one mask represents a detected target. The masks are de-projected into 3D points by matching with depth images, obtaining the 3D positions of the targets in the camera frame. Thereafter the coordinates are transformed from the camera frame to the robot arm frame based on camera extrinsic calibration. More details can be found in our previous work (Ge et al., 2019). The detection system outputs the 3D bounding boxes of the target strawberries and the thresholded point cloud for further obstacle calculation, as shown in Fig. 9(d). The obstacles around the target is determined based on the method described in Section 2.1. To calculate the number of points in each block, we crop the bounding box of each RoI block in point cloud using the *CropBox* function in the Point Cloud Library (PCL). Fig. 9(d) shows the obstacle bounding boxes (white) around a target, where only blocks occupied with obstacles are displayed.

3.2. Field test setup

As shown in Fig. 10, the field tests were performed on our newly developed U-shaped strawberry-harvesting robot (NORONN, www.

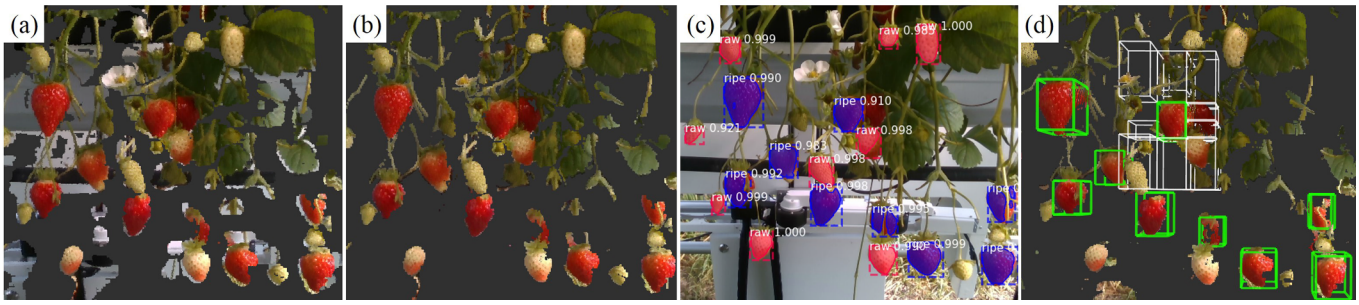

Fig. 9. The workflow of the fruit detection and obstacle determination: (a) original point cloud captured by an RGB-D camera; (b) 3D HSI color thresholding to remove the adjacent noise points from the background; (c) using deep learning to detect the ripe strawberries in an RGB image; (d) fruit localization and obstacle determination in 3D point cloud.

noronn.com). The picking system (Fig. 10(b)), including the camera, manipulators and LED panels, was mounted inside of the U-shaped arch. The robot passes though the strawberry table, covering the entire plants, with the possibility to use multiple manipulators picking on each side of the table. The new U-shaped architecture could prevent the vision system from being affected by the changing ambient illumination (sunlight), thus avoiding one of the main challenges facing the vision system in our old robot (Xiong et al., 2020). Different from our previous robot that used Cartesian arms, we developed a new low-cost SCARA-like robotic arm for the new harvesting robot. The 3 degrees of freedom arm has two rotation joints and one linear vertical axis using the same motors and control strategy as the old robotic arm (Xiong et al., 2020). The new arm is lighter and moving faster than the old Cartesian arms.

We conducted two sets of experiments in two places: a greenhouse at the Boxford Suffolk Farms (England) and a university experimental tunnel at the Norwegian University of Life Sciences (Norway). The tests were carried out on strawberry cultivars of “Malling Centenary” in England and “Murano” in Norway. The different biological characteristics may result in different performance of the robot.

3.3. Results

3.3.1. Application examples

Fig. 11 demonstrates three examples of using the proposed obstacle separation method in different situations. Fig. 11(a) and (b) show the cases testing on “Malling Centenary” and Fig. 11(c) illustrates an example on “Murano”.

In Fig. 11(a), a green obstacle is located above the target, which may be swallowed when the gripper is open to capture the target. Fig. 11(a-1) shows that there are no obstacles in layer 2, 3 and 4, but in layer 1, the central block $\boldsymbol{C}_{\mathrm{C}}$ and three other blocks $\boldsymbol{L}_{\mathrm{C}}$, $\boldsymbol{L}_{\mathrm{F}}$ and $\boldsymbol{C}_{\mathrm{F}}$ are occupied with obstacles, so based on Eq. (5), a drag operation is required. As there are no obstacles in layer 2, 3 and 4, the gripper moves up directly to enclose the target, as shown in Fig. 11(a-2) and (a-3). After holding the target, the gripper drags the target to the front-right region while moving upward where it contains fewer obstacles (Fig. 11(a-4)). At this position, if the gripper continues to move up, it might be difficult for the target to fall down towards a cutting position because the peduncle is inclined and the target has a contact force with the fingers. It may also damage the target with such a force. Therefore, in Fig. 11(a-5), the gripper pushes back to the central position while moving up for further fruit detachment (Fig. 11(a-6)), in which the upper obstacles are pushed aside. The blue line in Fig. 11(a-7) shows the recorded 3D trajectory of the gripper during the operation, while the red line is the trajectory projection on the xy plane, from which we can see that the gripper drags the target to $-y$ and $+x$ direction and then moves back. In the last image of each case, paths 1, 2 and 3 represent the three stages of operations in the bottom layer, central layers and top layer, respectively.

Fig. 11(b) and (c) demonstrate two more examples of the obstacle separation algorithm with more obstacles in the central layers and bottom layer. Only two blocks ($\boldsymbol{L}_{\mathrm{R}}$ and $\boldsymbol{C}_{\mathrm{R}}$) in layer 4 are unoccupied with obstacles for the target in Fig. 11(b-1). Therefore, the gripper uses the horizontal zig-zag push from the left rear to the right front (the red

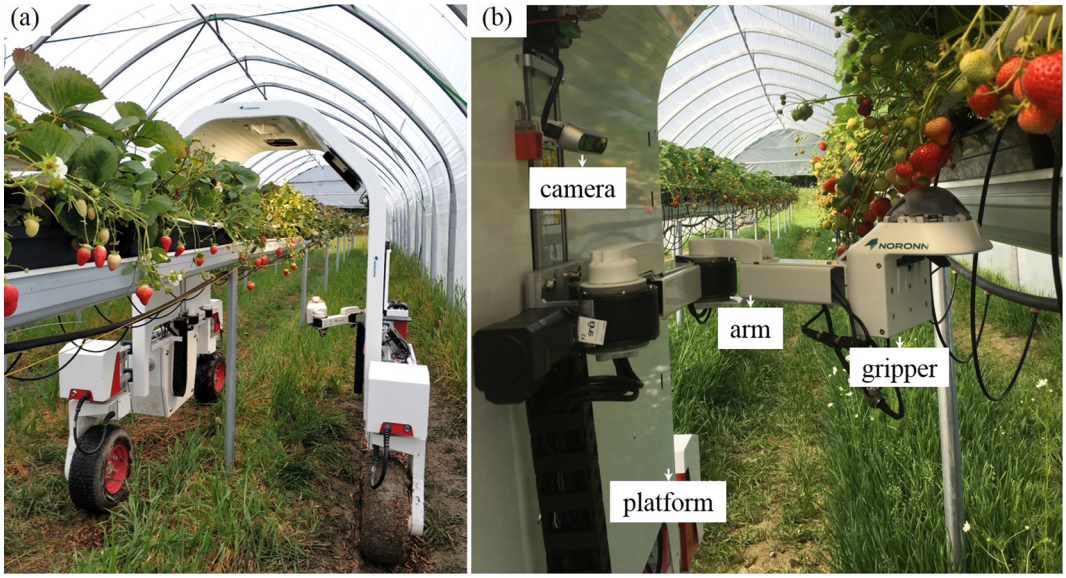

Fig. 10. The newly developed U-shaped strawberry-harvesting robot on a farm: (a) overview of the whole system; (b) inner view of the picking system.

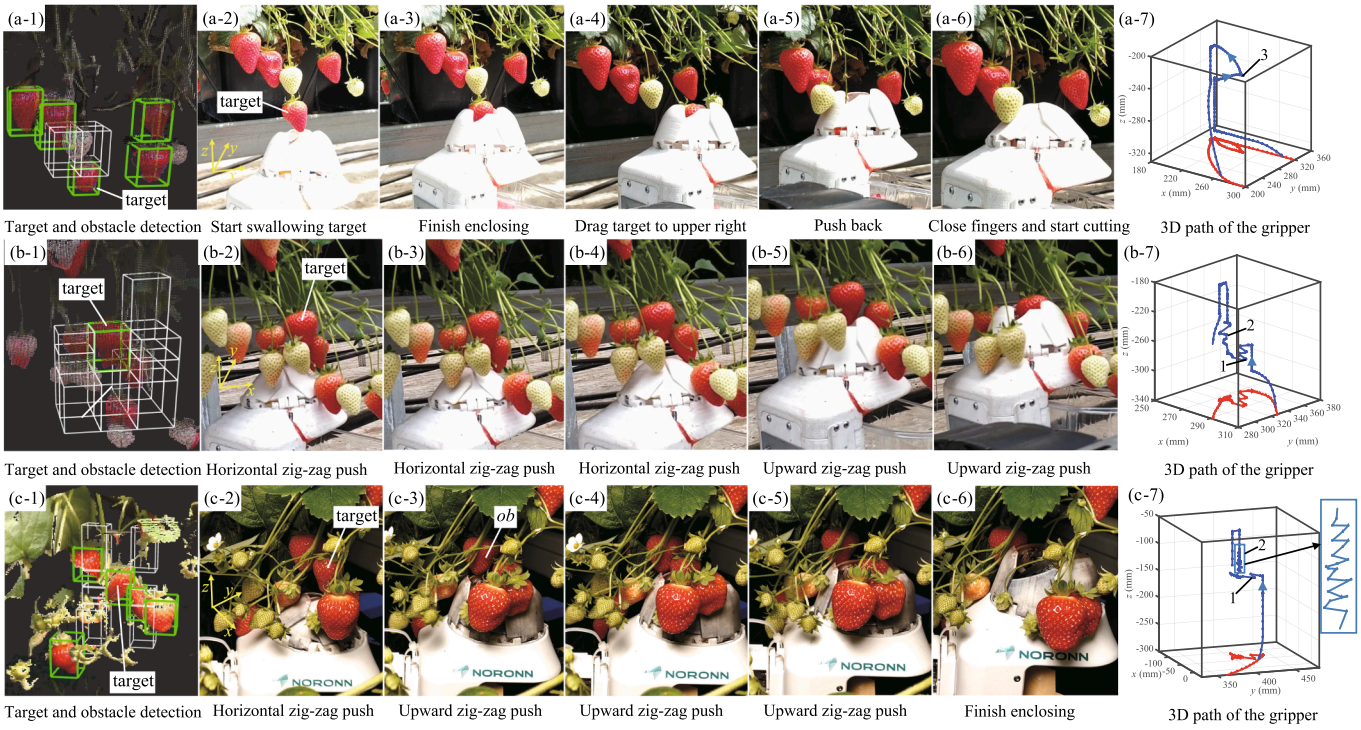

Fig. 11. Examples of the obstacle separation in the field test: each row of images represents a picking case, where the first (left) image shows the detection and localization of the target and obstacles and the last image (right) displays the 3D path of the gripper.

line in Fig. 11(b-7) shows the direction) to push the obstacles side to side, as can be seen in Fig. 11(b-2) to (b-4). Then, the gripper continues to use the upward zig-zag push operation to separate the obstacles in the central layers (Fig. 11(b-5) and (b-6)). Without this operation, the target and the obstacles may not be separated but pushed up together due to the contact force between each other. The situation in Fig. 11(c) is similar to Fig. 11(b). The system also determines to use the zig-zag push to clear the obstacles in the bottom layer and the central layers. Particularly, as shown in Fig. 11 (c-3), a red obstacle namely *ob* that is located in the upper left corner of the target may be captured by the gripper if it moves up directly. This may result in failure cutting of the target and also damage to the obstacle *ob*. With the upward zig-zag push, the gripper moves to the right side where the obstacle is out of the fingers (Fig. 11(c-4)). Then it moves back to the left and successfully pushes the obstacle *ob* aside (Fig. 11(c-5)).

3.3.2. Performance tests

The tests in England only include the results using the obstacle separation method, while in Norway, we conducted a comparison test, using and without using the obstacle separation method. For the tests in England, we did not include the closed-loop positional error control during picking (see details in (Xiong et al., 2019)), so it may increase the damage. While for the tests in Norway, the closed-loop control has been added before starting the operation in the central layers, which may reduce the picking speed compared to the England tests. For each setting, we implemented 100 attempts for the detected targets. Also, as the focus in this study is the obstacle separation method, only the targets with obstacles were used for the tests. This may result in a lower success rate compared to our previous reports, because the robot has a good performance on isolated strawberries (Xiong et al., 2020).

Fig. 12 reports the success rates in each stage and also the whole process under different settings. The whole process means the manipulation in the whole three stages: the bottom layer, central layers and top layer. In each independent stage, we only considered the results with obstacles in the corresponding layers, while a whole process may contain zero obstacle in one or two stages but at least one obstacle in all

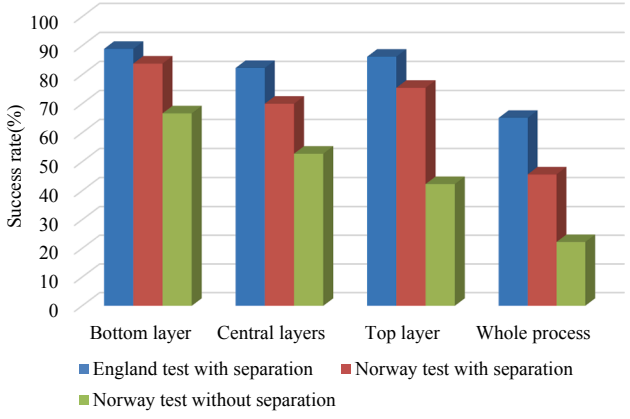

Fig. 12. Comparison of the success rates in different layers under different settings.

the layers. The success in the whole process means that in all the stages the separation is successful. Generally, the comparison tests in Norway show that the obstacle separation method is effective compared to the attempts without using the obstacle separation method. Also, the variety of "Malling Centenary" tends to be easier to be picked compared to "Murano". To be more precise, the comparison tests show that the drag operation in the top layer is most effective, increasing the success rate from 42.3% to 75.5% in the Norway tests. The bottom layer is the easiest layer in terms of obstacle separation. This might be attributed to the gripper design, since the opening action of the fingers under the target can help to push the obstacles aside. The manipulation in the central layers is difficult, showing a success rate of 70% and 52.8% with and without using the upward zig-zag push method, respectively. The success rate of the whole process is relatively low compared to the operation in the individual layer. One reason is that the success of the whole process means that all the three stages are successful. Also, we included the failures caused by the inaccurate localization of the targets

Table 1
The average number of obstacles and manipulation time.

Test settings	Average number of obstacles								Average manipulation time (s)	
	Bottom layer		Central layers		Top layer		Whole process			
Success cases	Failure cases	Success cases	Failure cases	Success cases	Failure cases	Success cases	Failure cases	Success cases	Failure cases	
England test with separation	2.4	3.0	2.0	2.7	1.8	2.0	5.4	8.0	5.0	4.7
Norway test with separation	4.2	4.4	3.1	4.0	2.7	2.6	7.6	8.8	7.6	6.9
Norway test without separation	2.8	3.3	2.4	2.5	2.4	2.4	4.8	5.9	5.3	4.8

before the manipulation start to the success rate calculation of the whole process, which takes up about 18% of all failures. It is also worth noting that in the whole process tests, one or two layers may have no obstacles. Therefore, the success rate of the whole process is not simply obtained by the product of the success rates of the three stages. For the test on “Murano”, the success rate of the whole process increases from 22.2% to 45.6% by using the separation method. The same separation method shows a better performance (65.1%) on the variety of “Malling Centenary”.

We also recorded the average number of obstacles and manipulation time for both success and failure cases, as shown in Table 1. The manipulation time only includes the picking time and arm traveling time (Xiong et al., 2020). Generally, the average number of obstacles in the success cases is fewer than that in the failure cases. This means that more obstacles will reduce the success rate. For the comparison tests in Norway, it is clear that without using the separation method, the average number of obstacles is fewer than using the separation method, which means that the separation method is able to work in the cases with more obstacles. In addition, the England tests have fewer obstacles in all cases, which indicates that “Malling Centenary” contains fewer clusters. As for the manipulation time, it can be seen that the success cases took more time than the failure cases. This is because that in the failures cases, the gripper may return once no targets are captured by the gripper based on the gripper internal sensing (Xiong et al., 2020). It is also evident that the tests using the separation method take about 2.2 s more than those without separation in the Norway tests.

3.3.3. Failure cases and discussions

As shown in Table 2, the most common failure reason before manipulation is the inaccurate depth sensing due to the front obstacles. Also, the control loop between the gripper and the arm is closed for positional error adjustment, while the control loop between the vision system and the manipulator is open (Xiong et al., 2020), so the predetermined target position on the first image frame might be changed when picking other berries. This is more considerable for “Malling Centenary” as the peduncles are longer and easier to swing after shaking. For the bottom layer, the most frequent failure is caused by the horizontal push, because the gripper might move the target together with other obstacles when pushing. This is easier to see when the target and obstacles grow on the same stem. Moreover, if the tip of the target berry is not detected, the top of the bottom layer is higher than the target. Then, the gripper might push directly on the target while moving other obstacles. On the other side, if the top of the bottom layer is lower than the target, the gripper might not be able to push the obstacles aside, resulting in swallowing obstacles.

For the central layers, the biggest issue for “Murano” is the mis-capturing of small obstacles due to the unsuccessful detection or too small obstacles to separate. This issue is not very frequent to “Malling Centenary”, since it does not have too many small flowers or berries growing on the same stem with the target. A noticeable failure issue for both varieties is that the gripper may get stuck due to the mis-capturing of other obstacles together with the target. Then, both the target and obstacles may be pushed up and could not be captured. One reason for this issue is the upward zig-zag push capturing obstacles occasionally. Also, the gripper may swallow obstacles and the target from the beginning due to the inaccurate localization. Another issue is that the gripper may not be able to separate connected berries due to the static contact force between them or short peduncles of the berries.

In the top layer, the most frequent failure for “Murano” is the case where a drag operation is not performed but required. This may be due to the unsuccessful detection of small objects or mis-capturing obstacles in other blocks when the central block in the top layer has no obstacles. However, for “Malling Centenary”, the common failure case is that the obstacles may move with the target during the drag operation due to the long peduncles or the insufficient separation due to short drag distance.

Learned from the failure cases, we think that a closed-loop vision guided manipulation system may improve the performance of the obstacle separation method considerably, in which the positions of the obstacles and the target are updated continuously. Considering the

Table 2
Failure reasons for the tests using the proposed separation method.

Stages	Failure reasons	Murano	Malling Centenary
Before manipulation	1. Target position was changed after picking other berries	18.7%	33.7%
	2. Inaccurate localization due to front obstacles	81.3%	66.3%
Bottom layer	1. Changed the target position when pushing obstacles	50%	56.3%
	2. Inaccurate estimation of the target bottom position, resulting in pushing the target away or swallowing obstacles	25%	31.2%
	3. Pushed new obstacles to the below of the target, resulting in swallowing small obstacles or pushing the target up	16.7%	12.5%
	4. Did not detect the small obstacles, so did not remove it before the operation in central layers	8.3%	0
Central layers	1. Swallowed small obstacles, due to failure detection or too small to separate	52.6%	38.9%
	2. Swallowed the target, but the connecting obstacles stopped the separation due to the static contact force or short peduncles	26.4%	16.7%
	3. One or more obstacles were captured together with the target, so the gripper is getting stuck	21%	44.4%
Top layer	1. Did not detect small obstacles, or swallowed obstacles in other blocks; did not drag when it is necessary	60.9%	21%
	2. Obstacles moving with the target while dragging or dragging distance is too short	26.1%	63.2%
	3. Dragged to a place that contains obstacles	13%	15.8%

operation speed, a combination of the currently used look-and-move method and a closed-loop refinement may lead to higher harvesting accuracy, and not so much decrease in the operation speed. Also, integrating one of the well-developed obstacle avoidance methods for the path planning prior to harvesting may also avoid some failures. In addition, the push and drag motions may increase the damage to other ripe fruits. Our future work will consider quantifying the damage rate and using soft material on the gripper fingers to reduce damage. Also, the thresholds N for determining the presence or absence of obstacles in blocks were manually tuned during the field tests. We will perform a statistical test to obtain more accurate values. It is also clear that "Malling Centenary" berries are easier to be picked due to the biological characteristics, such as more sparse clusters, long and independent peduncles, while "Murano" berries have more dense clusters with short peduncles growing on one stem. We suggest that more strawberry cultivars similar to "Malling Centenary" but with longer harvesting period and fewer clusters at one time could be bred to speed up the harvesting automation.

4. Conclusions

This paper presents a more sophisticated active obstacle separation strategy by introducing a combination of push and drag motions, which is an improvement to our previously proposed linear pushing policy. Different from the traditional obstacle avoidance methods, the proposed active obstacle separation method could actively separate obstacles from the target based on 3D visual perception. We show the separation policies, calculation methods and the image processing pipeline to implement the method on a newly developed robot. In addition to the old single linear push, a zig-zag push operation that consists of several linear pushes was proposed to separate the obstacles under and at the same height of the target, which is able to handle more dense obstacles because of the multi-directional pushes. Moreover, the generated side-to-side motion can break the static contact force between the target and obstacles, thus making it easier for the gripper to receive the target. Furthermore, we propose a novel drag operation to address the issue of mis-capturing obstacles located above the target, in which the gripper drags the target to a place with fewer obstacles and then pushes back to move the obstacles aside for further detachment. The operation includes avoiding obstacles and actively pushing aside obstacles. To implement the proposed method on a harvesting robot, an image processing pipeline was developed that uses 3D color thresholding to remove noise points from the background, 2D object detection based on deep learning to identify and localize the target and 3D point cloud operation to determine the obstacles. Field tests showed that the proposed method can improve the picking performance substantially. The performance may be further improved by incorporating a closed-loop vision guided manipulation system. A video of the field experiments can be found at https://drive.google.com/file/d/15BO2_4aaR5KHxbgOJQ9fV76zf0_i4GJs/view?usp=sharing.

CRediT authorship contribution statement

Ya Xiong: Conceptualization, Methodology, Software, Validation, Writing - original draft, Writing - review & editing. **Yuanyue Ge:** Data curation, Writing - review & editing, Visualization. **Pål Johan From:** Supervision, Writing - review & editing, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by Research council of Norway, FORNY2020, project number 2962020. We thank Mr. Robert England and Mr. Mihail Marita from the Boxford Suffolk Farms (Colchester, United Kingdom) for providing the strawberry greenhouse to conduct field experiments.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.compag.2020.105397.

References

Botterill, T., Paulin, S., Green, R., Williams, S., Lin, J., Saxton, V., Mills, S., Chen, X., Corbett-Davies, S., 2017. A robot system for pruning grape vines. J. Field Robot. 34 (6), 1100–1122. https://doi.org/10.1002/rob.21680.

Cao, X., Zou, X., Jia, C., Chen, M., Zeng, Z., 2019. Rrt-based path planning for an intelligent litchi-picking manipulator. Comput. Electron. Agric. 156, 105–118. https://doi.org/10.1016/j.compag.2018.10.031.

Danielczuk, M., Mahler, J., Correa, C., Goldberg, K., 2018. Linear push policies to increase grasp access for robot bin picking. In: 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE). IEEE, pp. 1249–1256. https://doi.org/10.1109/COASE.2018.8560406.

Dogar, M.R., Hsiao, K., Ciocarlie, M., Srinivasa, S.S., 2013. Physics-based grasp planning through clutter. Robotics: Science and Systems VIII, page 57. <https://ieeexplore.ieee.org/document/6578004>.

Ge, Y., Xiong, Y., Tenorio, G.L., From, P.J., 2019. Fruit localization and environment perception for strawberry harvesting robots. IEEE Access 7, 147642–147652. https://doi.org/10.1109/ACCESS.2019.2946369.

Kitaev, N., Mordatch, I., Patil, S., Abbeel, P., 2015. Physics-based trajectory optimization for grasping in cluttered environments. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 3102–3109. https://doi.org/10.1109/ICRA.2015.7139625.

Laskey, M., Lee, J., Chuck, C., Gealy, D., Hsieh, W., Pokorny, F.T., Dragan, A.D., Goldberg, K., 2016. Robot grasping in clutter: using a hierarchy of supervisors for learning from demonstrations. In: 2016 IEEE International Conference on Automation Science and Engineering (CASE). IEEE, pp. 827–834. https://doi.org/10.1109/COASE.2016.7743488.

Lehnert, C., Tsai, D., Eriksson, A., McCool, C., 2018. 3d move to see: Multi-perspective visual servoing for improving object views with semantic segmentation. arXiv preprint arXiv:1809.07896. https://arxiv.org/abs/1809.07896.

Moll, M., Kavraki, L., Rosell, J., et al., 2017. Randomized physics-based motion planning for grasping in cluttered and uncertain environments. IEEE Robot. Autom. Lett. 3 (2), 712–719. https://doi.org/10.1109/LRA.2017.2783445.

Van Henten, E.J., Hemming, J., Van Tuijl, B., Kornet, J., Meuleman, J., Bontsema, J., Van Os, E., 2002. An autonomous robot for harvesting cucumbers in greenhouses. Auton. Robots 13 (3), 241–258. https://doi.org/10.1023/A:1020568125418.

Xiong, Y., Ge, Y., Grimstad, L., From, P.J., 2020. An autonomous strawberry-harvesting robot: design, development, integration, and field evaluation. J. Field Robot. 37 (2), 202–224. https://doi.org/10.1002/rob.21889.

Xiong, Y., Peng, C., Grimstad, L., From, P.J., Isler, V., 2019. Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper. Comput. Electron. Agric. 157, 392–402. https://doi.org/10.1016/j.compag.2019.01.009.

Yaguchi, H., Nagahama, K., Hasegawa, T., Inaba, M., 2016. Development of an autonomous tomato harvesting robot with rotational plucking gripper. In: Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on. IEEE, pp. 652–657. https://doi.org/10.1109/IROS.2016.7759122.

Yamamoto, S., Hayashi, S., Yoshida, H., Kobayashi, K., 2014. Development of a stationary robotic strawberry harvester with a picking mechanism that approaches the target fruit from below. Japan Agric. Res. Quart.: JARQ 48 (3), 261–269. https://doi.org/10.6090/jarq.48.261.

Zeng, A., Song, S., Welker, S., Lee, J., Rodriguez, A., Funkhouser, T., 2018. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 4238–4245. https://doi.org/10.1109/IROS.2018.8593986.